

University of Groningen

Geometrically projected discrete dislocation dynamics

Akhondzadeh, Hamed; Sills, Ryan; Papanikolaou, Stefanos; van der Giessen, Erik; Cai, Wei

Published in:
Modelling and Simulation in Materials Science and Engineering

DOI:
[10.1088/1361-651X/aacf31](https://doi.org/10.1088/1361-651X/aacf31)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Final author's version (accepted by publisher, after peer review)

Publication date:
2018

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Akhondzadeh, H., Sills, R., Papanikolaou, S., van der Giessen, E., & Cai, W. (2018). Geometrically projected discrete dislocation dynamics. *Modelling and Simulation in Materials Science and Engineering*, 26(6), [065011]. <https://doi.org/10.1088/1361-651X/aacf31>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Submitted to: MSMSE

Geometrically projected discrete dislocation dynamics

Sh. Akhondzadeh^{a,b}, R. B. Sills^c, S. Papanikolaou^{d,e}, E. Van der Giessen^f, and W.
Cai^{a,*}

^a Department of Mechanical Engineering, Stanford University, Stanford CA 94305, USA

^b Department of Civil Engineering, Johns Hopkins University, Baltimore, MD 21218, USA

^c Sandia National Laboratories, Livermore, CA 94551, USA

^d Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA

^e Departments of Mechanical Engineering and Physics, West Virginia University, Morgantown, WV, 26506, USA

^f Zernike Institute for Advanced Materials, University of Groningen, Groningen 9747 AG, The Netherlands

Abstract

Three-dimensional discrete dislocation dynamics methods (3D-DDD) have been developed to explicitly track the motion of individual dislocations under applied stress. At present, these methods are limited to plastic strains of about one percent or less due to high computational cost associated with the interactions between large numbers of dislocations. This limitation motivates the construction of minimalistic approaches to efficiently simulate the motion of dislocations for higher strains and longer time scales. In the present study, we propose Geometrically Projected Discrete Dislocation Dynamics (GP-DDD), a method in which dislocation loops are modeled as

* Corresponding author.
E-mail address: Caiwei@stanford.edu

geometrical objects that maintain their shape with a constant number of degrees of freedom as they expand. We present an example where rectangles composed of two screw and two edge dislocation segments are used for modeling gliding dislocation loops. We use this model to simulate single-slip loading of copper and compare the results with detailed 3D-DDD simulations. We discuss the regimes in which GP-DDD is able to adequately capture the variation of the flow stress with strain rate in the single-slip loading condition. A simulation using GP-DDD requires ~ 40 times fewer degrees of freedom for a copper single slip-loading case, thus reducing computational time and complexity.

Keywords: dislocation dynamics, Frank-Read source, single-slip, plasticity

1. Introduction

Three-dimensional discrete dislocation dynamics (3D-DDD) is a method for simulating crystal plasticity, at a length scale of micrometers [1-2]. 3D-DDD simulations have the potential to provide great insight into the physics of dislocation-mediated plasticity, since dislocation-scale processes such as junction formation and cross-slip can be directly accounted for. As with all coarse-grained simulation methods, 3D-DDD depends on local mechanisms that are typically elucidated by lower length-scale models, such as molecular dynamics. However, even when molecular mechanisms are well understood, decades of research and algorithm development [3] have shown that 3D-DDD simulations are still computationally very expensive: achieving strains of order 1% in a tensile test typically requires thousands of CPU hours [1, 2, 4].

In an effort to elucidate collective effects of dislocation plasticity on mechanical properties, a variety of alternatives to 3D-DDD have been developed. A major focus has been on the development of continuum dislocation dynamics (CDD) models [5-7], within which dislocations are represented in terms of density tensor formulations [8]. The benefit of CDD is that the computational cost of simulations does not scale with the total dislocation density. However, it is unclear which governing principle or degrees of freedom should be taken into account for constructing the coarse-grained equations of motion.

On the other hand, 2D simulations of ensembles of discrete, straight edge dislocations have helped us gain insight into the complexity of dislocation plasticity [9]. Based on this minimal set of

dislocations, there have been several discrete dislocation plasticity models that are designed for loading geometries and conditions that preserve this minimal dislocation ensemble, such as in thin films or mode-I crack tips, forming the so-called two-dimensional discrete dislocation dynamics methods (2D-DDD) [10]. In these models, dislocations are treated as infinitely long and parallel dislocations that may propagate through the material without curving. Additional mechanisms that may originate when dislocations become curved, such as junction formation, may be included in the 2D model through effective considerations that give rise to the so-called 2.5-dimensional-DDD method [12-13]. In this formulation, dislocations are still approximated as infinitely long parallel lines, but with dislocations on different glide planes able to “collide” with each other and form junctions. However, the constraints imposed by the two-dimensional nature of these minimal models restrict their applications. In this paper, we interpret these 2D-DDD models as examples of the more general geometrically projected discrete dislocation dynamics. We propose a natural generalization of the 2D models to incorporate 3D effects in a coarse-grained dislocation dynamics model.

The complexity of dislocation plasticity as viewed through such 2D-DDD or 2.5D-DDD methods [11-13] as well as experiments at the nanoscale [13-14] has been demonstrated to show non-trivial size effects [15], collective spatiotemporal aspects [16], and far-from equilibrium, abrupt characteristic behaviors [17-18]. Motivated by the statistical mechanics of non-equilibrium complex systems, it is expected that such non-equilibrium collective effects require coarse-grained dynamical approaches to adequately understand the multiple inter-connected space and time scales.

In this work, we present a novel minimal approach to dislocation dynamics which is three dimensional, yet significantly less computationally expensive than 3D-DDD. We call this new approach Geometrically Projected Discrete Dislocation Dynamics (GP-DDD)². In GP-DDD we study the motion of dislocation loops, and in order to reduce the number of degrees of freedom in the simulation we project, at each time step, each loop onto a pre-selected geometric form. In the simplest version, each loop is projected onto a rectangle composed of two screw and two edge dislocation segments as shown in Fig. 1(a). The major feature of this method is the control of the dislocation ensemble topology in a way that it remains simple. Here, we focus on the use of

² An earlier version of the GP-DDD model was presented in [30]

rectangular projections, but the approach can readily be extended to higher order projections (pentagons, hexagons, etc.). The principle difference between GP-DDD and traditional 3D-DDD is that “remeshing,” the set of rules whereby additional degrees of freedom are introduced to bowing dislocation lines, is suppressed in order to keep the dislocation shape simple. In this way, the number of degrees of freedom only increases as the number of dislocation loops increases. The principle objective of this work is to present the GP-DDD method through the rectangle projection example, demonstrate that it is considerably less expensive than 3D-DDD (~40 times fewer degrees of freedom) yet displays reasonable agreement with 3D-DDD in simple single-slip loading.

The remainder of this manuscript is organized as follows. In Section 2.1 a brief overview of the 3D-DDD procedure is presented. Section 2.2 explains the GP-DDD method, including dislocation loop generation and collision rules. Simulations of single slip loading of copper using the GP-DDD method are presented in section 3 and they are compared with 3D-DDD results obtained with the *ParaDiS* code. Section 4 discusses possible routes on how to extend the GP-DDD method beyond single slip-loading followed by a summary in section 5.

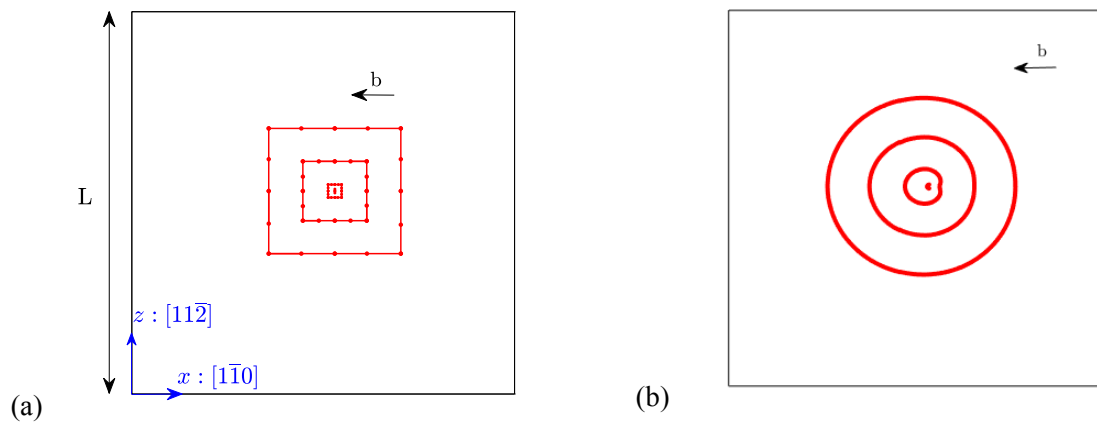


Figure 1. Snapshots of (a) GP-DDD with rectangular loops each discretized into 16 segments and (b) 3D-DDD simulation at $t = 2.29 \mu\text{s}$ of a single $3.5 \mu\text{m}$ source on a single plane in a large cell ($L = 350 \mu\text{m}$) subjected to $\dot{\sigma}_{xy} = 5 \times 10^6 \text{ MPa.s}^{-1}$. Each side of the rectangular loop in GP-DDD is discretized into four segments to keep the maximum segment length below a cut-off length required by the force calculation implementation.

2. Methods

2.1. Review of three-dimensional discrete dislocation dynamics

In 3D-DDD simulations [2,19] each dislocation line is discretized into a finite number of segments connected by nodes, and the location of these nodes is updated at each time step. The discretization is modified according to remeshing rules, usually based on maximum and minimum segment lengths, l_{max} and l_{min} [1].

Each time step of a node-based 3D-DDD method proceeds as follows [2]. First, the force on each segment is computed (we discuss the details of force calculation below), and then distributed to the connecting nodes to obtain nodal forces. Next, using a mobility law the nodal velocities are determined, from which displacements can be obtained by time integration to update the location of the nodes. The corresponding plastic strain increment is then calculated, collisions between segments are detected, and junction formation or dissociation events are handled. Lastly, dislocation segments are remeshed, and all of the above operations are repeated at the next time step.

The contributions to the force on a dislocation segment include applied external stress f^{app} , core energy f^{core} , and elastic interactions with itself as well as other dislocation segments f^{el} , i.e., $f^{total} = f^{app} + f^{core} + f^{el}$. The calculation of forces due to elastic interactions with other segments is the most computationally expensive part of 3D-DDD simulation. If elastic interactions between all of the segment-segment pairs within the simulation cell are calculated explicitly, the cost scales as $O(N^2)$ where N is the number of segments. However, the number of operations can be reduced by approximating the forces between two-well separated segments using the fast multipole method (FMM) [1], which reduces the computational cost to $O(N)$.

Another important issue in 3D-DDD is time integration. The type of integration scheme used to obtain the new nodal positions at the next time step has a great impact on the speed of the simulation. In order to maximize the efficiency of our 3D-DDD simulations, we employ the subcycling-based time integrator [20] for our *ParaDiS* simulations. We did not develop an equivalent subcycling-based integrator for the new GP-DDD code, however, and used the Heun integrator instead. Hence, in order to make direct performance comparisons with GP-DDD, we

also ran a few *ParaDiS* simulations using the Heun method without subcycling (the default integrator in *ParaDiS*).

2.2. Geometrically projected discrete dislocation dynamics

The GP-DDD approach features three types of objects: dislocation loops, Frank-Read (FR) sources and shapes resulting from self-collision of dislocation loops due to periodic boundary conditions. The major concerns with 3D-DDD involve the drastic increase of computational cost and memory usage at large dislocation densities. GP-DDD reduces the computational cost by coarse graining the discretization of the dislocation network in such a way that as each dislocation loop expands, its total number of degrees of freedom does not increase. GP-DDD is based on projecting the dislocation network onto a discrete set of shapes that are pre-selected and are consistent with the included dislocation mechanisms. During the progression of the dislocation network, these shapes can expand or combine according to the allowed dislocation mechanisms in the simulation, without generating any shape other than the pre-selected ones.

Below we present the details of the GP-DDD algorithm. Most of the steps in a GP-DDD simulation are very similar to 3D-DDD, except that evolution of the dislocation structure is constrained to contain only a set of allowed shapes. A major difference between the two approaches is that for GP-DDD special Frank-Read source objects need to be developed. In 3D-DDD, sources can be introduced by simply pinning the ends of a short dislocation line or they may form naturally as the network evolves. This line then bows out and emits a loop according to the classical Frank-Read mechanism. In GP-DDD, we approximate Frank-Read sources as straight dislocation segments that are not allowed to bow out, but generate loops according to constitutive rules defined in the next section.

2.2.1. Frank-Read (FR) sources

In GP-DDD simulations, we need a source to generate loops. For this purpose, we introduce FR sources with length L_{FR} , activation stress τ_{act} , and loop nucleation time $t_{nuc}(\tau)$, where τ is the local shear stress at the FR source. When the shear stress acting on the FR source exceeds τ_{act} , the source becomes unstable and bows out until a dislocation loop is nucleated. In GP-DDD, we emulate this process by representing each FR source as a straight dislocation line of finite length,

and then introduce new dislocation loops when $\tau > \tau_{act}$ and $t > t_{nuc}(\tau)$ for that source, where t is the simulation time that has elapsed since the last nucleation event by that source. The calculation details of τ_{act} and $t_{nuc}(\tau)$ are explained in the Appendix A. In simulations where the stress state varies with time (e.g., when a constant strain rate is imposed), the nucleation time $t_{nuc}(\tau)$ varies as well. To account for this effect, the parameter \mathcal{R} is introduced as

$$\mathcal{R} = \sum_j \frac{\Delta t_j}{t_{nuc}(\tau_j)} \quad (1)$$

where $\tau_j > \tau_{act}$ is the shear stress on the FR source at time step j and Δt_j is the magnitude of the time step during step j . The GP-DDD criterion for adding a loop around a FR source in the varying stress state case is $\mathcal{R} \geq 1$. As soon as a loop is added around a source, \mathcal{R} is reset to zero. This approximation is expected to work well if the local stress on the FR source varies slowly with time.

2.2.2. Force computation

In GP-DDD, we compute driving forces on dislocation segments in the same way as with 3D-DDD. However, the expansion rate of a rectangular dislocation loop (i.e., rate at which area is swept out by the loop) is different from a true dislocation loop of the same area. In order to retain the same plastic strain rate (hence same expansion rate) in the projection process, we introduce “force multipliers.” Each force contribution (applied stress, elastic energy, core energy) requires a different force multiplier. A derivation for these multipliers is presented in Appendix B by comparing the expansion of idealized circular and square dislocation loop. We obtain multipliers of $\frac{\sqrt{\pi}}{2}$, $0.912(\frac{\pi}{2})$, and 0.7843 for the applied stress, elastic energy, and core energy contributions, respectively. These multipliers ensure that the area-sweep-rate (which is proportional to the plastic strain rate) of a single, isolated square loop is the same as that of a circular loop. In FCC metals, the dislocation mobility does not have a strong dependence on character (e.g. edge or screw), so that the aspect ratio of dislocation loops are not significantly different from 1. As a result, the multipliers derived by comparing circles and squares provide a reasonably good approximation here. However, if the loops are very elongated (such as dislocations in BCC metals), then different multipliers may be needed.

It must be noted that these multipliers are only valid in the case of rectangular dislocation loops, and new multipliers should be derived if other GP-DDD shapes, such as hexagons, are used.

2.2.3. Collision of segments

Similar to many 3D-DDD codes, in GP-DDD two dislocation segments with opposite signs are said to have collided when their minimum separation distance is less than the annihilation radius r_{ann} . However, since only certain geometric shapes are allowed in GP-DDD, we have to handle collisions in special ways in order to avoid introducing disallowed shapes into the simulation domain. The main collision geometries of interest for the single slip simulations we will present here, where only a single Burgers vector is considered and no forest dislocations are present, are as follows.

Two separate dislocation loops collide with each other

In this case, the two loops are merged together into a new loop, as shown in Fig. 2. The area of the new loop is the sum of the areas of the old loops. The centroid of the new loop is selected as the average of the positions of the colliding loop centroids weighted by their area. The loop aspect ratio is equal to the aspect ratio of the smallest rectangle bounding the colliding loops.

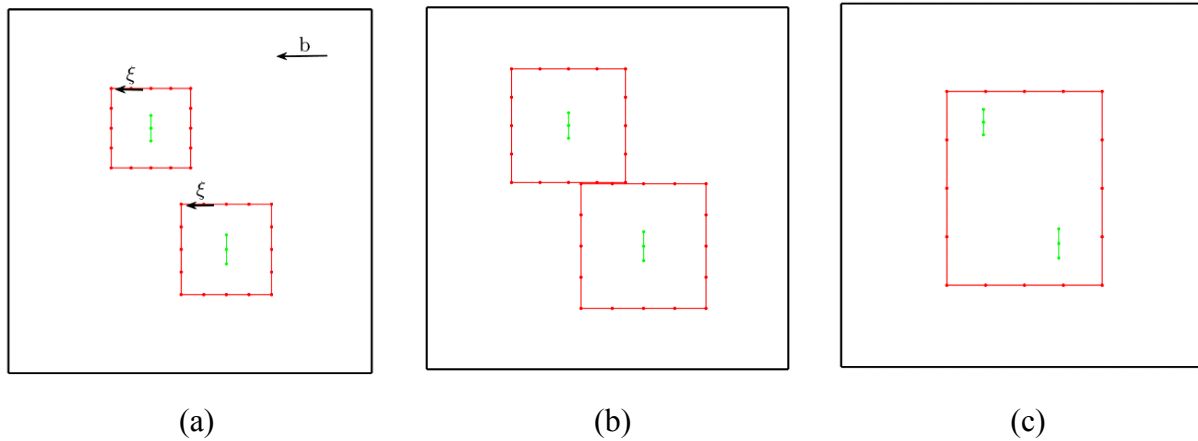


Figure 2. Merging of two separate glide loops, which have been discretized into 16 segments, having the same sign: (a) two loops added around two sources (FR segments shown in green) (b) two loops collide and the collided segments have opposite signs (c) two collided loops merge together and conserve the total area.

A dislocation loop collides with a FR source segment.

For two FR sources denoted by A and B on the same plane, it is possible that due to time lag between their loop nucleation criteria, one of them (say, A) nucleates a loop, denoted by ℓ^A , which expands and reaches the other source B when \mathcal{R}^B for the source B , as defined in Eq. (1), is less than 1. The loop nucleation criteria for the B has not been met and there is no loop around B . Thus, loop ℓ^A collides with the source B . If B were an actual FR source as simulated in 3D-DDD, it would be at some intermediate stage of loop nucleation, which may be described by a value of the nucleation parameter, \mathcal{R}^B , between zero and one. Collision of loop ℓ^A with the partially nucleated source B results in loop ℓ^A accumulating the area already swept by source B and source B resetting completely. Hence, in order to be consistent with the behavior of FR sources in 3D-DDD, the following steps are taken when the loop ℓ^A collides with the source B :

- i. loop ℓ^A expands by area $\mathcal{R}^B \times A_i$ to take into account the developed plastic strain of the source B before it nucleates a loop. A_i is the initial area of a loop to be added around the source as explained in Appendix A.
- ii. \mathcal{R}^B for the source B is reset to zero.

Self-collision of a dislocation loop

Since we use periodic boundary conditions in our simulations, it is possible for a dislocation loop to collide with itself across a periodic boundary. In these cases, the colliding loop segments annihilate with each other, resulting in either a pair of isolated, straight segments or total annihilation of the loop.

3. Results

3.1. Computational set up

GP-DDD simulations were performed by modifying the MATLAB code *DDLab*³, to be consistent with the GP-DDD algorithm as presented above. For comparison, 3D-DDD simulations were performed using *ParaDiS*. Periodic boundary conditions in all directions were used and long range

³ Serial version of the code *ParaDiS* available at <http://micro.stanford.edu/~caiwei/Forum/2005-12-05-DDLab>

forces were calculated using a precomputed table [21]. Material properties used are as shown in Table 1 along with other relevant parameters.

For all simulations only one slip system is active, thus cross-slip and multiple slip are not considered. A cubic simulation cell with edges of length L was used. The three repeat vectors of the simulation cell are chosen to be along the $x = [1\bar{1}0]$, $y = [111]$ and $z = [11\bar{2}]$ directions, and only the slip system $[1\bar{1}0]$ -(111) is considered. Hence, the slip plane in the simulation frame is the x - z plane (cf. Fig 1) and segments on one plane exiting a periodic boundary re-enter the cell from the opposing side on the same plane.

TABLE 1. SIMULATION PARAMETERS

Property	Parameter	Value
Shear Modulus	μ	54.6 GPa
Poisson's ration	ν	0.324
Burgers vector magnitude	b	0.255 nm
Drag coefficient	B	15.6 $\mu\text{Pa} \cdot \text{s}$
Annihilation radius	r_{ann}	0.1 μm (GP-DDD) ⁴
		15 b (3D – DDD)

We consider test cases consisting of edge FR sources distributed on either a single plane or multiple parallel planes. All dislocation segments are confined to move in their predefined glide planes. Furthermore, no forest dislocations (dislocations piercing glide planes) are present. Burgers vectors of all sources as well as all the added loops are $\vec{b} = \frac{b}{\sqrt{2}}[\bar{1}10]$ in the crystal frame, which is

⁴ A large annihilation radius was necessary in order to not miss collisions with GP-DDD since time steps were so large [20]. With ParaDiS, time steps were much smaller allowing for a smaller annihilation radius of $0.3825 \times 10^{-2} \mu\text{m}$. We have confirmed that the difference in r_{ann} did not affect the major findings in this work.

in the negative x -direction in the simulation frame. The simulation results are presented in the next section, starting with single plane followed by multiple planes.

3.2. Single plane results

Firstly, in order to validate the loop nucleation and force calculation procedures, a single FR source of length $3.50\ \mu\text{m}$ in a large simulation cell ($L = 350\mu\text{m}$) subjected to a constant shear stress rate of $\dot{\sigma}_{xy} = 5 \times 10^6\ \text{MPa.s}^{-1}$ was simulated using GP-DDD and 3D-DDD methods. We consider a constant shear stress rate loading condition because it provides a simple test case where the stress state varies in time. Snapshots of GP-DDD and 3D-DDD simulations at the same instances of time are presented in Figure 1. The area swept by dislocations vs. time is plotted in Fig. 3(a). The vertical dashed lines in this figure denote the loop nucleation times. The GP-DDD and 3D-DDD results are quite similar. Since a number of the procedures used in GP-DDD are approximate (force multipliers assume circular loops and uniform stress state, the loop nucleation model is based on a constant, uniform stress state, etc.), we do not expect an exact match between the two simulations.

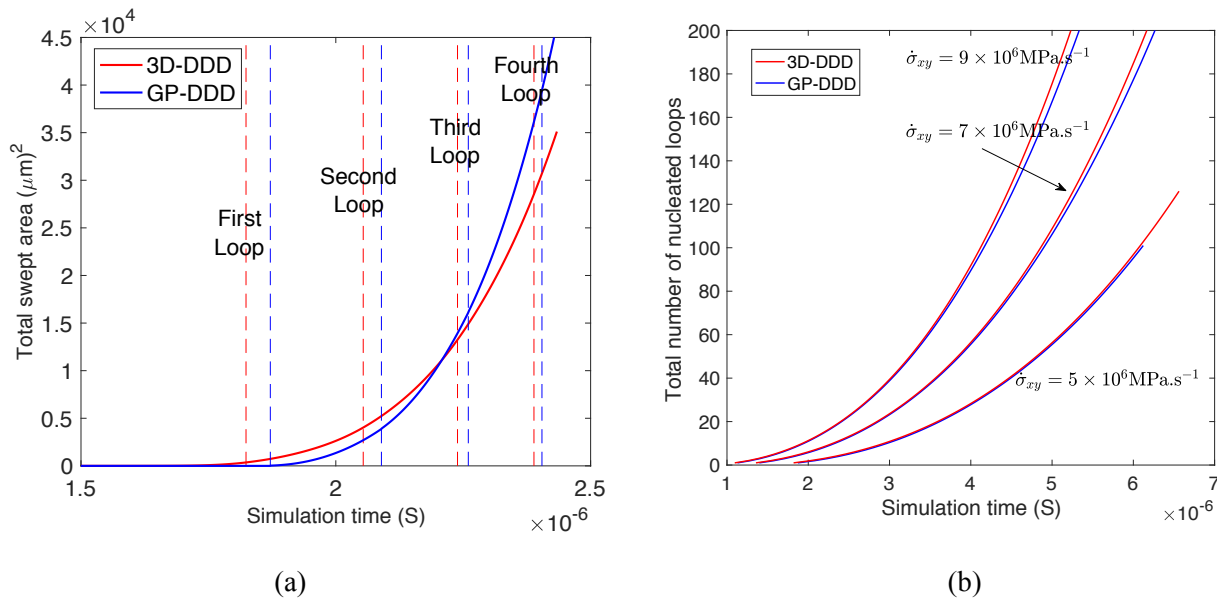


Figure 3. Results for a single $3.5\mu\text{m}$ FR source under a constant shear stress rate $\dot{\sigma}_{xy}$ (a) Comparison of total swept area by dislocations in a large cell ($L = 350\mu\text{m}$) subjected to $\dot{\sigma}_{xy} = 5 \times 10^6\ \text{MPa.s}^{-1}$.

Vertical lines denote loop nucleation events. (b) Comparison of total number of nucleated loops vs. time in a cell size with $L = 35\mu m$ subjected to different constant shear stress rates.

To further validate the GP-DDD procedures in a case where a greater number of loops is emitted, the size of the simulation cell was reduced to $L = 35\mu m$. The same simulations were performed subjected to three different values of shear stress rate. The total number of nucleated dislocation loops as a function of time is plotted in Fig. 3(b) for each case (because of annihilation across periodic boundaries, at most only three loops are present in the cell at a time, however). Again, GP-DDD and *ParaDiS* are in good agreement.

3.3. Multiple plane results

In order to show the performance of GP-DDD with a more complex test case, two configurations were selected and simulated: six parallel planes each containing one source, and three planes with two sources on each, which are denoted by S_1P_6 and S_2P_3 , respectively. Both cases have six sources: two of length $3.5\mu m$, two of length $4\mu m$, and two of length $4.5\mu m$. These simulations are intended to emulate the single slip response of bulk, single crystalline copper, with the primary output being the shear stress-shear strain curve. Hence, we imposed a constant shear strain rate $\dot{\gamma}$ instead of a constant shear stress rate. At each time step, the stress state is updated based on the plastic and total strain increments.

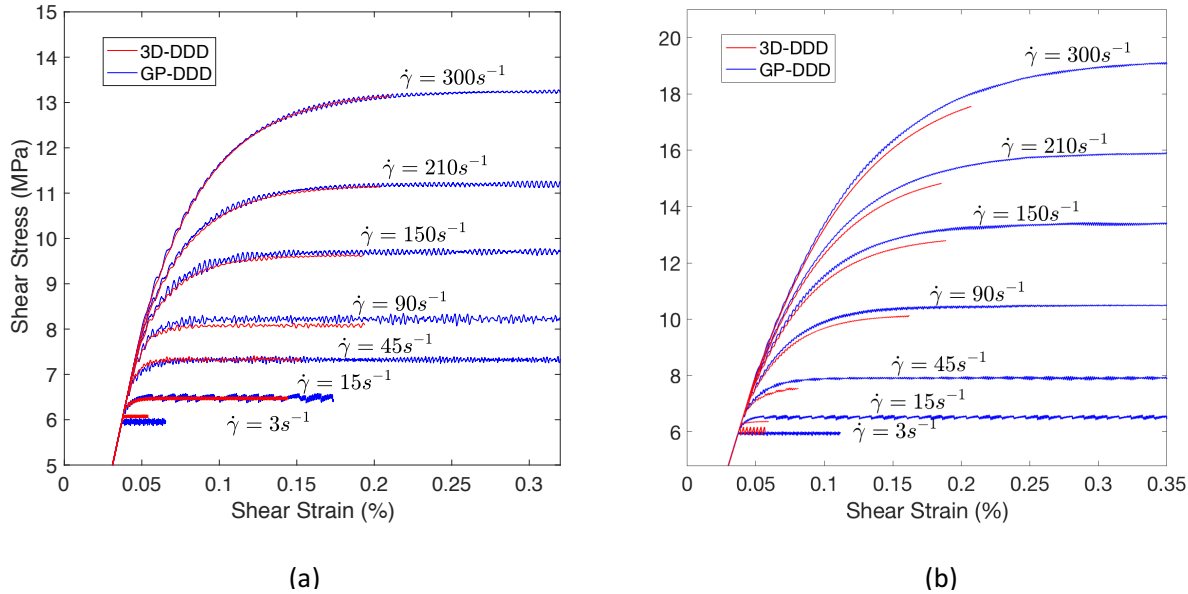


Figure 4. Comparison of shear stress-strain curves; simulations of S_1P_6 (a) and S_2P_3 (b) subjected to different values of constant shear strain rate in a $50\mu m$ simulation box. S_1P_6 consists of six planes, with one FR sources on each of them while S_2P_3 includes three planes, with two FR sources on each of them. 3D-DDD results correspond to discretization length of $l_{max} \approx (1/10) L_{FR}$.

Fig. 4(a) and Fig. 4(b) presents the evolution of the stress-strain curve for the S_1P_6 and S_2P_3 cases, respectively, with a $50\mu m$ simulation cell subjected to different shear strain rates ranging from 3 to $300 s^{-1}$. The distance between planes was $8\mu m$ and $20\mu m$ for S_1P_6 and S_2P_3 , respectively. It can be seen that there is very good agreement between both methods, which demonstrates that the essential physics of the problem is captured by the GP-DDD approximation. A pair of snapshots from GP-DDD and *ParaDiS* at a shear strain of 0.1% are presented in Fig. 5 showing very similar microstructures. We also note that GP-DDD is able to attain much higher strains at all strain rates, despite the fact that the code was written in MATLAB, run in serial (as opposed to 8 CPUs with *ParaDiS*), and does not use an efficient subcycling integrator. No hardening was observed in any GP-DDD or 3D-DDD simulations; instead, each asymptoted to a well-defined flow stress dependent upon the source density and shear strain rate.

To further quantify the comparison between GP-DDD and 3D-DDD and validity of the loop-source collision approximation as explained in the section 2.2.3, S_2P_3 simulations were repeated for different cell sizes L , and the 0.1%-offset yield strengths, τ_Y , were extracted. The obtained yield strengths are presented in the Table 2. We provide results for shear strain rates of 90, 150,

210 and 300 s^{-1} , and provide the average percent error between GP-DDD and 3D-DDD for each box size. The maximum error observed is about 5% with an average error of 2.3%.

Table 2. Yield strengths (0.1%-offset) of S_2P_3 cases corresponding to different simulation box sizes. 3D-DDD simulation were performed using $l_{\max} = 1500b$

	$L = 40\mu m$		$L = 50\mu m$		$L = 60\mu m$	
	GP-DDD	3D-DDD	GP-DDD	3D-DDD	GP-DDD	3D-DDD
$\dot{\gamma} = 90$	9.1	8.9	9.9	9.6	10.1	10.6
$\dot{\gamma} = 150$	10.8	10.5	11.6	11.3	12	12.4
$\dot{\gamma} = 210$	11.8	11.6	12.5	12.3	12.5	13
$\dot{\gamma} = 300$	12.7	12.6	13.4	13.2	13.7	13.7
Avg Error	1.8 %		2.2 %		2.9 %	

Finally, in Fig. 6 we plot the yield stresses obtained with GP-DDD in the S_1P_6 and S_2P_3 configurations for shear strain rates ranging from 0.3 to 300 s^{-1} . We note that results with $\dot{\gamma} < 3 \text{ s}^{-1}$ could not be obtained with *ParaDiS* due to unreasonably long simulation times. Hence, GP-DDD provides access to simulation conditions that are out of reach with 3D-DDD. We find that for the considered simulation geometries, the quasi-static limit (yield strength independent of strain rate) is attained for shear strain rates of 3 s^{-1} or lower.

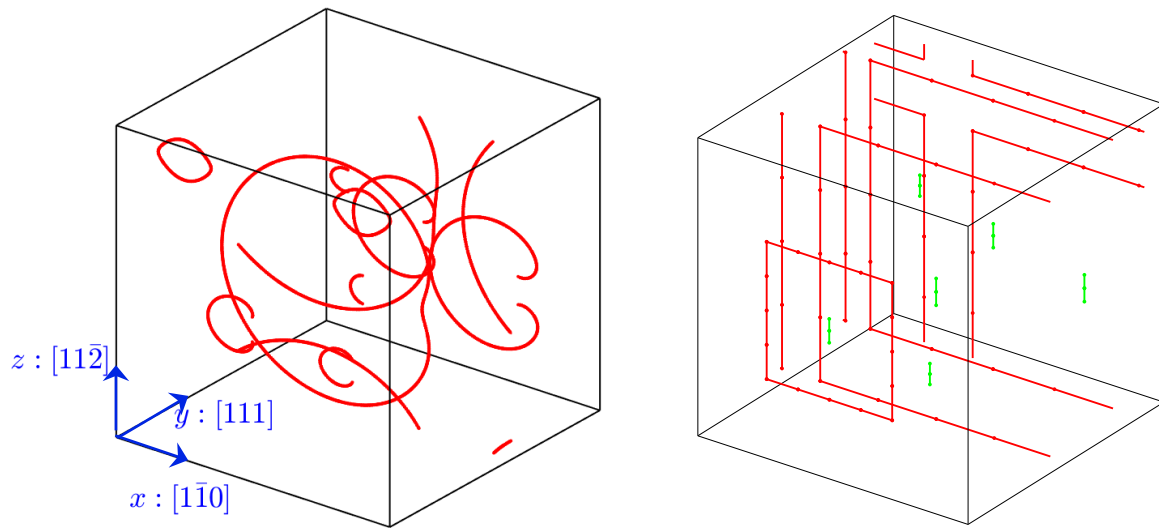


Figure 5. Snapshots of 3D-DDD and GP-DDD simulations of S_1P_6 problem in a $50\mu m$ box subjected to $\dot{\gamma} = 150 \text{ s}^{-1}$ at the same overall shear strain of 0.1%.

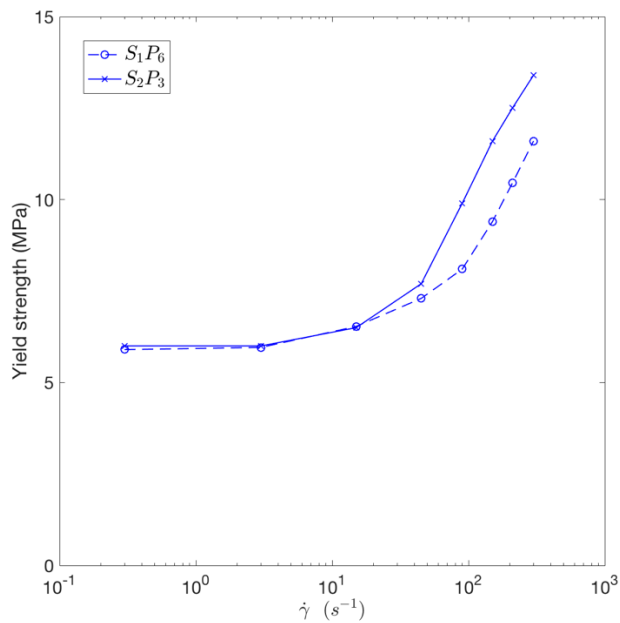


Figure 6. Yield stresses (0.1%-offset) S_1P_6 and S_2P_3 cases obtained by GP-DDD simulations as functions of the shear strain rate.

3.4 Performance comparison

In order to compare the performance of the codes in terms of computational cost, we directly use a GP-DDD configuration from 0.3% shear strain as an initial condition and simulate a small amount of strain using GP-DDD and *ParaDiS* with the Heun integrator. Since GP-DDD is implemented in MATLAB and *ParaDiS* is implemented in C/C++, the wall times between *ParaDiS* and GP-DDD cannot be compared directly⁵. Instead, we compare the accumulative number of segment-segment force calculations, N_{ss} , executed by each code over a small amount of strain. To clarify, if the time integrator converges in its first iteration, N_{ss} calculated over time interval $t_2 - t_1$ is $N_{ss} = \sum_{t=t_1}^{t_2} \frac{n_t(n_t-1)}{2}$ where n_t denotes the current number of degrees of freedom.

The slope $dN_{ss}/d\gamma$ is a measure of the computational cost for a simulation to reach a unit amount of strain. We find $dN_{ss}/d\gamma \approx 10^{15}$ and 10^{10} for *ParaDiS* and GP-DDD, respectively. This indicates that GP-DDD is about one hundred-thousand times faster than *ParaDiS*. There are two key reasons for this significant performance gap. The first is that GP-DDD has far fewer degrees of freedom (segments). Fig. 7(a) shows the number of degrees of freedom as a function of strain for the S₁P₆ case with a shear strain rate of 300 s^{-1} . GP-DDD requires about 40 times fewer degrees of freedom. The second reason is that because the dynamics is greatly simplified in GP-DDD, destabilizing modes that reduce the time step are suppressed, enabling a larger time step to be taken [22]. Fig. 7(b) shows how the time step varies with strain between the two approaches. With GP-DDD, the time step is limited so that the nodal displacements during each time step do not exceed the annihilation radius (which would result in a missed collision), whereas in *ParaDiS* the time step varies wildly dependent upon the error during time integration induced by destabilizing modes (e.g., a FR source annihilating). The results indicate a time step that is about 100 times larger with GP-DDD. Combining these two efficiency gains leads to a speed up of about $40^2 \cdot 100 \approx 10^5$, consistent with the $dN_{ss}/d\gamma$ values obtained above.

⁵ For reference, the results in Fig. 4 with GP-DDD required about 24 hours on a single CPU (MATLAB code) and with *ParaDiS* about 300 hours on 8 CPUs (C/C++ code).

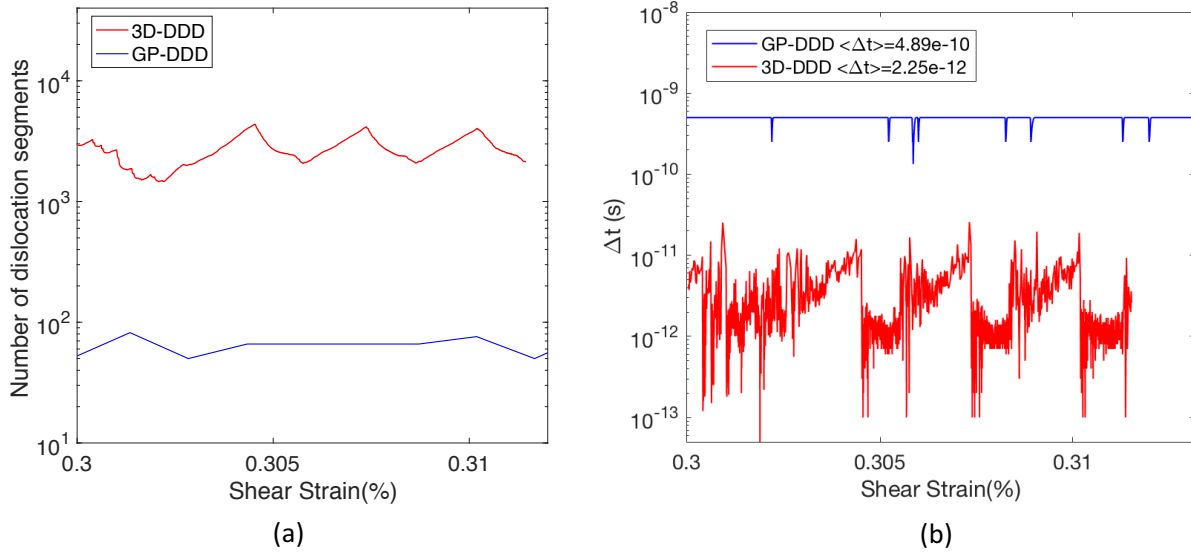


Figure 7. Comparison of results corresponding to S_1P_6 simulations subjected to $\dot{\gamma} = 300 \text{ s}^{-1}$ in a box size of $50\mu\text{m}$ using the Heun integrator; the initial configuration was chosen as GP-DDD results at 0.3% shear strain: (a) number of degrees of freedom (b) time step size

4. Discussion

We have presented a new coarse-grained discrete dislocation dynamics method wherein dislocation loops are projected onto a set of pre-determined shapes, such as rectangles or hexagons. Performing this projection significantly reduces the number of degrees of freedom (dislocation segments) necessary to represent the loops, resulting in tremendous efficiency gain. These projected loops are able to move, merge and annihilate based on rules derived from the physics of dislocation dynamics. This approach is similar to 2D and 2.5D-DDD in that it is a coarse-grained, minimal model of dislocation-mediated plasticity. However, GP-DDD is three-dimensional in nature and is able to naturally capture the increase in dislocation density and plastic strain generation due to an expanding dislocation loop. Hence, the GP-DDD approach provides a closer connection between basic dislocation physics and mechanical response.

In the present paper, rectangles were selected as the GP-DDD shapes. Generalizing the approach to other shapes require relatively minor modifications to the code. For example, the rules governing loop nucleation, expansion, and collision are independent of loop geometry. However,

the multipliers that take into account the expansion rate change due to the projection (see Appendix B) are dependent upon the shape type.

We focused on single slip simulation of copper for which there are no forest dislocations and dislocations are confined to move in pre-defined glide planes. In this case, the relevant physics is nucleation and expansion of dislocation loops from FR sources. By carefully focusing on these processes (see Appendix A and B), we were able to develop a GP-DDD model that is largely consistent with the more rigorous 3D-DDD model. However, it is important to note that GP-DDD is not a tool intended for studying the fine-scale details of arbitrary dislocation structure evolution; by its very nature as a coarse-grained model, it is designed to study large-scale ensemble behaviors of the system in regimes for which it is calibrated. Hence, any physical phenomenon dominated by fine-scale processes is not likely to be captured well by GP-DDD, at least not without careful consideration and potential modification to the algorithm.

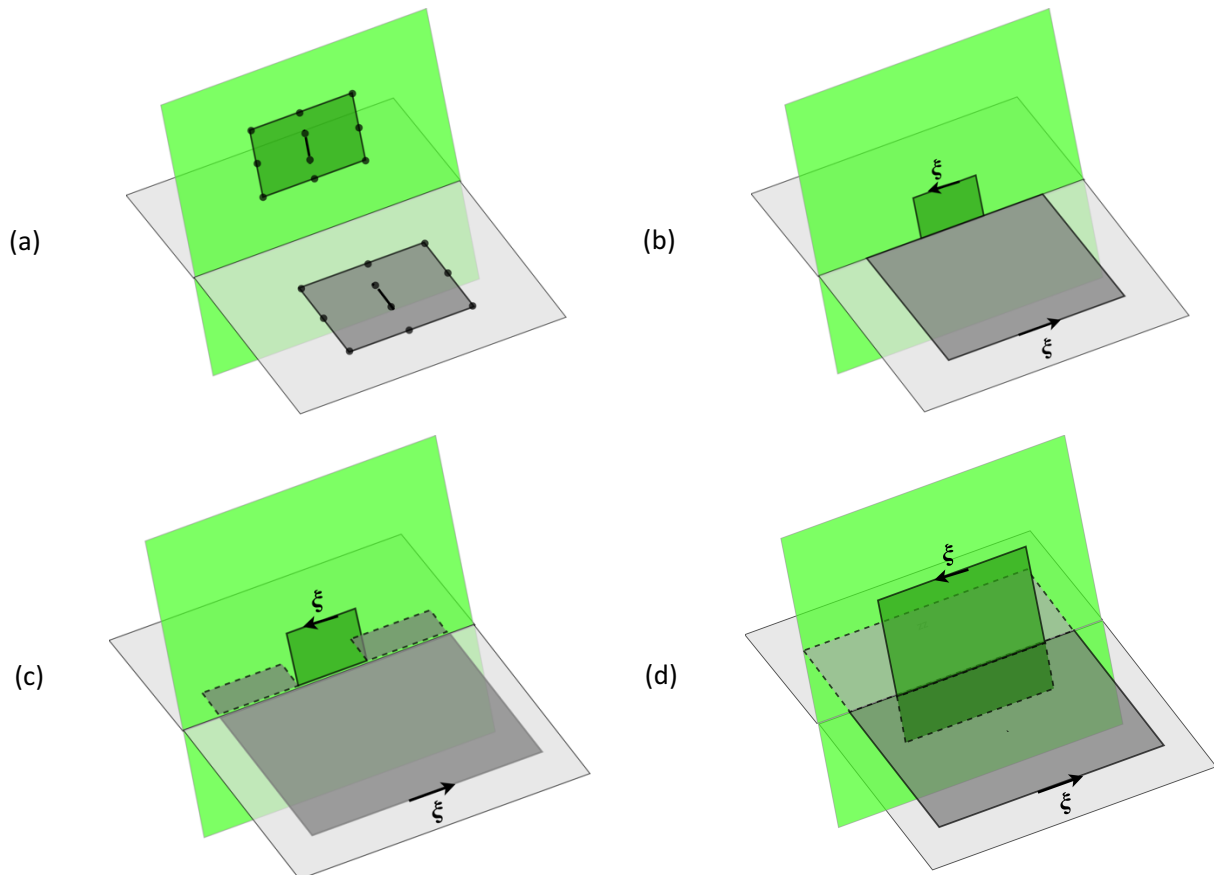


Figure 8. Possible extensions of GP-DDD. (a) Multiple slip-planes by introducing sources on different planes and defining a physical law at forest intersection. (b, c, d) Cross-slip by introducing a smaller loop with the same Burgers vector on the second plane. The segments in the new and original loops at the intersection of the two planes cancel each other. The other three segments on the new loop represent the cross-slipped dislocation. In 3D-DDD, cross-slip can produce temporary FR sources on both the original and cross-slipped plane. (c) We can emulate the activation of the temporary FR source in the original plane by adding smaller loops with the same Burgers vector on the original plane. (d) At a later stage, these smaller loops can merge with each other and then with the larger loop, producing a single loop on the original plane. This also liberates the loop on the cross-slip plane to expand freely.

While we have focused on single slip in this paper, the presented approach can be easily extended to multiple slip simulations by applying the same methods to another slip plane and defining a new rule to handle junction formation at forest intersections (see figure 8(a)). When two, non-coplanar (forest) dislocation lines collide, they may react and form a junction dislocation. These junctions are important because they contribute to the strength and strain hardening of the dislocation network [23-24]. One potential approach for incorporating junctions in the GP-DDD framework is to define special “junction nodes” that form at the forest intersection points and glue the two lines together until a criterion for breaking the junction is satisfied. For example, we could define a junction strength that is dependent upon the Burgers vectors and line directions of the dislocations forming the junction. Note that in this approach, the actual, finite-length junction dislocation is not explicitly represented. Instead, a zero-length junction node is introduced, whose strength is informed by 3D-DDD simulations of actual, finite-length junctions. Cross-slip—the process of a screw dislocation changing its glide plane—is another important physical process associated with multiple slip. In GP-DDD, a cross-slip event can be introduced by inserting a new dislocation loop with the same Burgers vector on the cross-slip plane as illustrated in figure 8(b). The new loop represents the cross-slipped section of dislocation line, which is able to bowout and form a new expanding loop in the cross-slip plane. The details of the process by which a dislocation cross-slips are complex and may produce a temporary Frank-Read source [25]. In Fig. 8(c-d) we show a possible scenario in which a cross-slip event may evolve and lead to two glissile dislocation loops in the GP-DDD model. The details of the cross-slip rules in the GP-DDD model will need to be designed to mimic the dislocation behavior in 3D-DDD simulations, in the same way that the rules for the Frank-Read source were designed in Section 2.2.1.

In all the examples considered, only one Burgers vectors on parallel slip planes were present, which allows rectangular loops of only screw and edge character with the same orientation. Thus,

rules to handle loop collision can be easily defined as explained in the section 2.2.3. In the case where different sources with different Burgers vectors are present, additional rules should be defined to handle collisions between loops having different Burgers vectors. Simulations could take one of the following forms. Each of these approaches should be further studied to evaluate its accuracy.

- i. For two sources S_1 and S_2 , with Burgers vectors b_1 and b_2 , introduce rectangular loops around each source that are aligned with its Burgers vector. This way, S_1 and S_2 will nucleate loops consisting of only screw and edge dislocations that have different orientations with respect to each other (for FCC materials, this orientation is 60°).
- ii. Alternatively, S_1 and S_2 can nucleate loops that are not aligned with their respective Burgers vector, but rather are aligned with only one of the b_1 or b_2 . This way, one of the loops consists of mixed dislocations. Implementation of cross-slip in this approach would be more challenging.
- iii. Choose a different loop shape that is able to more easily accommodate different Burgers vectors (e.g., triangles, hexagons).
- iv. Completely suppress collisions of loop with different Burgers vectors, namely by putting dislocation sources with different Burgers vector on different planes.

Due to its low computational cost, GP-DDD method could potentially be used for fatigue simulations with cyclic loading over many cycles, perhaps enabling a deeper understanding of persistent slip band (PSB) formation [26]. PSBs are often comprised of long, parallel bundles of dislocations. It has been experimentally shown that during fatigue loading dislocation segments span across adjacent PSBs [26], and this is believed to affect their growth. Simulation of these spanning dislocations is not possible with a 2D approach. However, such a configuration is possible using GP-DDD where the 3D loop structure is preserved.

Obstacles such as particles or precipitates can also be added to the GP-DDD simulation. In this case, one potential approach to handle loop and obstacle intersection is to hold the intersecting side of the dislocation in place until an obstacle strength has been overcome. A proper obstacle strength must be defined in order to decide when a dislocation loop passes an obstacle after intersecting with it. This obstacle strength will be dictated by the type of obstacle and its properties (e.g., incoherent spherical precipitate with a modulus mismatch). In low obstacle density cases,

dislocation loops can cross the obstacles one by one. Examples of this approach are present in our earlier study [29]. However, if the density of the obstacles is high, in reality the dislocation will start to bow out between obstacles; these bowing events can be emulated by adding smaller loops with the same Burgers vector as the original loop that remains glued to the obstacle as shown in the Fig. 9. Subsequent collision events may cause these new loops to recombine with the original loop, as shown in Fig. 9(c).

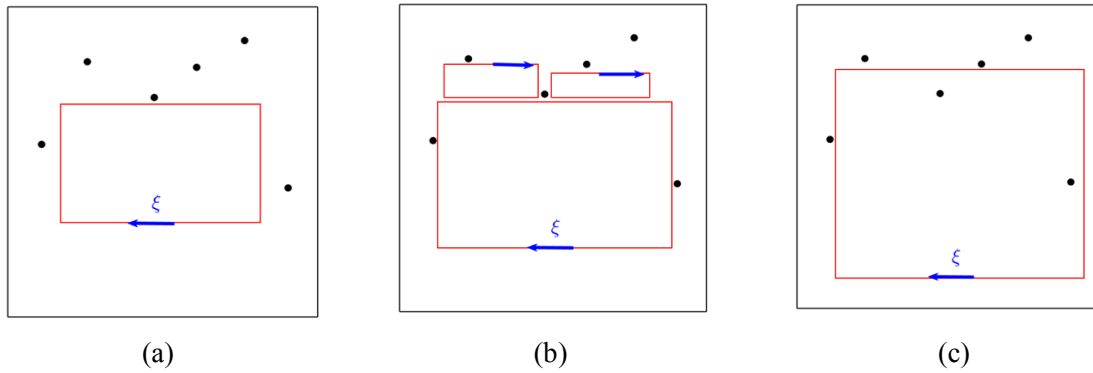


Figure 9. Dislocation loops intersecting with obstacles. (a) At first dislocation loop get stuck at the obstacles (shown by black dots) and motion of the loop past the obstacle is hindered until the obstacle strength has been overcome. (b) To emulate the process of dislocation bowing out, smaller loops with the same Burgers vector can be added between obstacles. (c) Whenever the original loop passes the initial obstacle, the smaller loop will be absorbed by the original loop.

A final point we wish to make is that because the degrees of freedom in GP-DDD and 3D-DDD are nearly identical, it is straightforward to move back and forth between the two approaches. Transitioning from GP-DDD simply requires inputting the node and segment information into a *ParaDiS* input file, including info about each segment's glide plane. *ParaDiS* automatically increases the number of degrees of freedom through its remeshing algorithm, subject to the user-defined remesh parameters. For example, Fig. 9 shows a 3D-DDD simulation with GP-DDD configuration used as initial configuration. In fact, we exploited this fact in the performance comparison presented in Section 3.4. This provides a potential path to studying plasticity at high strains with GP-DDD providing the high strain microstructures for use by 3D-DDD to study the mechanical behavior. Conversely a configuration from 3D-DDD, (after being decomposed into a superposition of loops) can be projected and simplified down to a GP-DDD configuration.

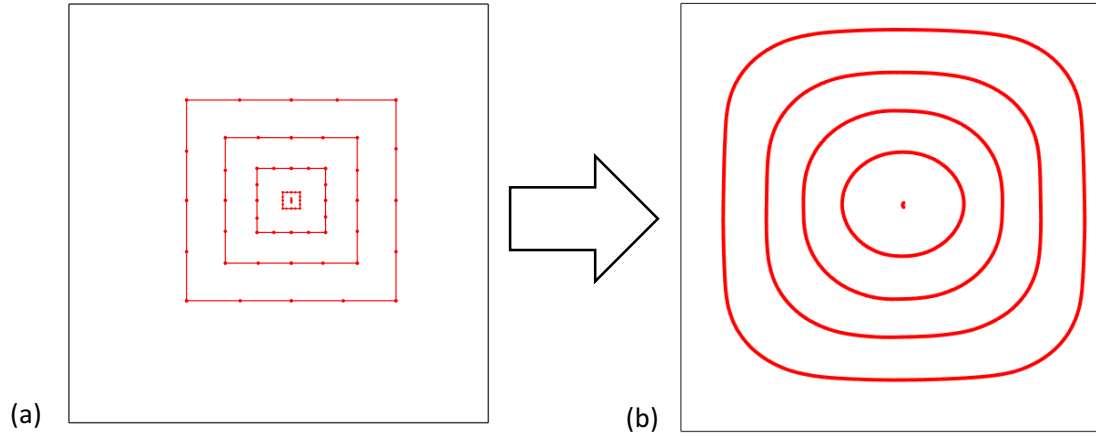


Figure 10. (a) GP-DDD configuration to be used as 3D-DDD initial configuration. (b) 3D-DDD simulation continued from the GP-DDD configuration (a).

5. Conclusion

A new, coarse-grained, three-dimensional, minimal approach to discrete dislocation dynamics was proposed in which dislocation loops are projected onto specific geometric shapes, such as rectangles. The approach consists of 1) developing a loop nucleation criterion based on 3D-DDD simulation results; 2) updating the positions of the projected loops by similar rules as in 3D-DDD with the only difference being the use of corrective force multipliers; and 3) developing special rules to handle collision between two coplanar loops and a coplanar loop with a FR source. Simulation of single slip loading of copper was performed using both GP-DDD and 3D-DDD and the comparison of stress-strain curves and number of segment-segment pair interactions were presented. The results indicate that GP-DDD significantly reduces the computational cost while producing results consistent with 3D-DDD simulation. For this reason, GP-DDD enables us to simulate FR sources subjected to lower strain rates than conventional 3D-DDD. We expect GP-DDD to be most suited for single-slip and fatigue simulations while conventional 3D-DDD is better suited for multi-slip and work hardening simulations. Lastly, even though we have focused on developing GP-DDD for single slip without obstacles, it can potentially be extended to multiple slip cases where obstacles are present and cross-slip occurs.

Acknowledgements

We would like to thank H. Song and D. M. Dimiduk for discussions. S.A. would also like to thank Dr. Nicolas Bertin for his guidance in running the 3D-DDD simulations. This work was initiated at a DOE PIs’ meeting for the “Mechanical Behavior and Irradiation Effects” program. We would like to acknowledge support from DOE under awards DE-SC0014109 (S.P.) and DE-SC0010412 (S.A. and W.C.). R.B.S. acknowledges Sandia National Laboratories. Sandia National Laboratories is a multitechnology laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525.

References

[1] A. Arsenlis *et al.*, “Enabling strain hardening simulations with dislocation dynamics,” *Model. Simul. Mater. Sci. Eng.*, vol. 15, no. 6, p. 553, 2007.

[2] V. Bulatov and W. Cai, *Computer simulations of dislocations*, vol. 3. Oxford University Press on Demand, 2006.

[3] L. Kubin, *Dislocations, mesoscale simulations and plastic flow*, vol. 5. Oxford University Press, 2013.

[4] B. Devincre and L. Kubin, “Scale transitions in crystal plasticity by dislocation dynamics simulations,” *Comptes Rendus Phys.*, vol. 11, no. 3–4, pp. 274–284, 2010.

[5] I. Groma, F. F. Csikor, and M. Zaiser, “Spatial correlations and higher-order gradient terms in a continuum description of dislocation dynamics,” *Acta Mater.*, vol. 51, no. 5, pp. 1271–1281, 2003.

[6] S. Limkumnerd and J. P. Sethna, “Mesoscale theory of grains and cells: crystal plasticity and coarsening,” *Phys. Rev. Lett.*, vol. 96, no. 9, p. 95503, 2006.

[7] A. Acharya and A. Roy, “Size effects and idealized dislocation microstructure at small scales: predictions of a phenomenological model of mesoscopic field dislocation

- mechanics: Part I,” *J. Mech. Phys. Solids*, vol. 54, no. 8, pp. 1687–1710, 2006.
- [8] L. D. Landau and E. M. Lifshitz, *Course of theoretical physics*. Pergamon Press, 1959.
- [9] R. J. Amodeo and N. M. Ghoniem, “Dislocation dynamics. I. A proposed methodology for deformation micromechanics,” *Phys. Rev. B*, vol. 41, no. 10, p. 6958, 1990.
- [10] E. Van der Giessen and A. Needleman, “Discrete dislocation plasticity: a simple planar model,” *Model. Simul. Mater. Sci. Eng.*, vol. 3, no. 5, p. 689, 1995.
- [11] A. A. Benzerga, Y. Bréchet, A. Needleman, and E. Van der Giessen, “Incorporating three-dimensional mechanisms into two-dimensional dislocation dynamics,” *Model. Simul. Mater. Sci. Eng.*, vol. 12, no. 1, p. 159, 2003.
- [12] D. Gomez-Garcia, B. Devincre, and L. P. Kubin, “Dislocation patterns and the similitude principle: 2.5 D mesoscale simulations,” *Phys. Rev. Lett.*, vol. 96, no. 12, p. 125503, 2006.
- [13] J. R. Greer and J. T. M. De Hosson, “Plasticity in small-sized metallic systems: Intrinsic versus extrinsic size effect,” *Prog. Mater. Sci.*, vol. 56, no. 6, pp. 654–724, 2011.
- [14] M. D. Uchic, P. A. Shade, and D. M. Dimiduk, “Plasticity of micrometer-scale single crystals in compression,” *Annu. Rev. Mater. Res.*, vol. 39, pp. 361–386, 2009.
- [15] M. D. Uchic, D. M. Dimiduk, J. N. Florando, and W. D. Nix, “Sample dimensions influence strength and crystal plasticity,” *Science (80-.)*, vol. 305, no. 5686, pp. 986–989, 2004.
- [16] S. Papanikolaou *et al.*, “Quasi-periodic events in crystal plasticity and the self-organized avalanche oscillator,” *Nature*, vol. 490, no. 7421, pp. 517–521, 2012.
- [17] S. Papanikolaou, Y. Cui, and N. Ghoniem, “Avalanches and Plastic Flow in Crystal Plasticity: An Overview,” *arXiv Prepr. arXiv1705.06843*, 2017.
- [18] S. Papanikolaou, H. Song, and E. Van der Giessen, “Obstacles and sources in dislocation dynamics: Strengthening and statistics of abrupt plastic events in nanopillar compression,” *J. Mech. Phys. Solids*, vol. 102, pp. 17–29, 2017.
- [19] L. P. Kubin, G. Canova, M. Condat, B. Devincre, V. Pontikis, and Y. Bréchet, “Dislocation microstructures and plastic flow: a 3D simulation,” in *Solid State*

- Phenomena*, 1992, vol. 23, pp. 455–472.
- [20] R. B. Sills, A. Aghaei, and W. Cai, “Advanced time integration algorithms for dislocation dynamics simulations of work hardening,” *Model. Simul. Mater. Sci. Eng.*, vol. 24, no. 4, p. 45019, 2016.
- [21] W. Cai, V. V Bulatov, J. Chang, J. Li, and S. Yip, “Periodic image effects in dislocation modelling,” *Philos. Mag.*, vol. 83, no. 5, pp. 539–567, 2003.
- [22] R. B. Sills and W. Cai, “Efficient time integration in dislocation dynamics,” *Model. Simul. Mater. Sci. Eng.*, vol. 22, no. 2, p. 25003, 2014.
- [23] R. Madec, B. Devincre, and L. P. Kubin, “From dislocation junctions to forest hardening,” *Phys. Rev. Lett.*, vol. 89, no. 25, p. 255508, 2002.
- [24] B. Devincre, T. Hoc, and L. Kubin, “Dislocation mean free paths and strain hardening of crystals,” *Science (80-.)*, vol. 320, no. 5884, pp. 1745–1748, 2008.
- [25] P. J. Jackson, “Dislocation modelling of shear in fcc crystals,” *Prog. Mater. Sci.*, vol. 29, no. 1–2, pp. 139–175, 1985.
- [26] C. Laird, P. Charsley, and H. Mughrabi, “Low energy dislocation structures produced by cyclic deformation,” *Mater. Sci. Eng.*, vol. 81, pp. 433–450, 1986.
- [27] A. J. E. Foreman, “The bowing of a dislocation segment,” *Philos. Mag.*, vol. 15, no. 137, pp. 1011–1021, 1967.
- [28] A. A. Benzerga, “An analysis of exhaustion hardening in micron-scale plasticity,” *Int. J. Plast.*, vol. 24, no. 7, pp. 1128–1157, 2008.
- [29] W. Cai, A. Arsenlis, C. R. Weinberger, and V. V Bulatov, “A non-singular continuum theory of dislocations,” *J. Mech. Phys. Solids*, vol. 54, no. 3, pp. 561–587, 2006.
- [30] S. Akhondzadeh, “Geometrically projected discrete dislocation dynamics.”, Master Thesis, Johns Hopkins University, 2016.

Appendix A. Loop nucleation model in GP-DDD

The loop nucleation model for GP-DDD is completely defined through activation stress τ_{act} and loop nucleation time $t_{nuc}(\tau)$. We obtain these two parameters by performing 3D-DDD simulations of a single edge FR source with different initial lengths under various amounts of applied shear stress.

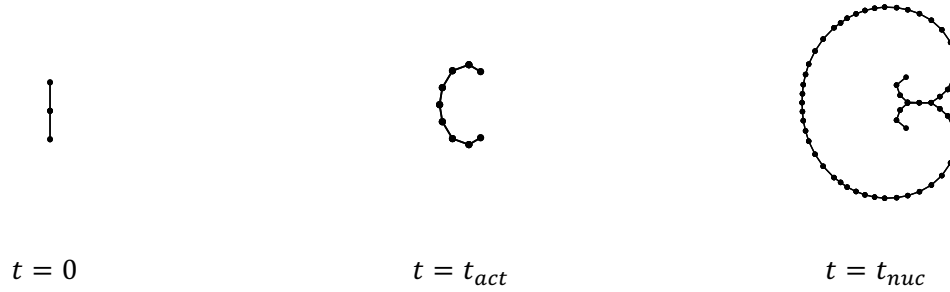


Figure A.1 – snapshots of initial FR source, activated FR source and FR source at time of loop emission in a 3D-DDD simulation.

$t_{nuc}(\tau)$ is the time required for the straight FR segment under constant shear stress τ , to nucleate a loop, and it can be written as the summation of two time intervals: $t_{act}(\tau)$ and $t_{nuc}(\tau) - t_{act}(\tau)$, which are the times required for the straight FR segment to reach its critical state and the time from the critical state to loop nucleation, respectively (see Fig. A.1). Foreman [27] and Benzerga [28] proposed the following expressions for τ_{act} and t_{act} ⁶:

$$\tau_{act} = A \frac{\mu b}{2\pi L_{FR}} \ln \left(\frac{L_{FR}}{r_c} \right) \quad (\text{A.1})$$

$$t_{act} = \frac{B h_c}{\tau b} \left(1 + \frac{2}{\xi} (I_1 + I_2) \right) \quad (\text{A.2})$$

⁶ Eq. (2) of [28] gives an expression for the critical nucleation time, which is identical to what we here call activation time t_{act}

where A is a fitting parameter close to unity, r_c is the dislocation core radius, B is the drag coefficient of the dislocation line, τ the applied shear stress on the FR segment, ξ is the ratio of applied shear stress to the activation stress $\xi = \frac{\tau}{\tau_{act}}$, h_c depends on the character of the initial segment ($h_c = 0.75L_{FR}$ and $h_c = 0.325L_{FR}$ for edge and screw segments, respectively) and I_1 and I_2 are defined as follows:

$$I_1 = \frac{1}{2} \ln(2 - 2/\xi)$$

$$I_2 = \frac{1}{\sqrt{\xi^2 - 1}} \left[\tan^{-1} \left(\frac{\xi - 1}{\sqrt{\xi^2 - 1}} \right) + \tan^{-1} \left(\frac{1}{\sqrt{\xi^2 - 1}} \right) \right]$$

For the simulation parameters used here, the value of A in Eq. (1) is found to be $A = 1.25$. We find that the nucleation time is dependent on the length of FR source L_{FR} , and the maximum discretization length l_{max} . In order to obtain a uniform definition for $t_{nuc}(\tau)$ which is independent of the L_{FR} , values of $t_{nuc}(\tau)$ are divided by $t_{act}(\tau)$ determined using Eq. (A.2); the results are plotted against ξ in Fig. A.2. Fig. A.2(a) shows the effect of l_{max} and Fig. A.2(b) elaborates the effect of L_{FR} . It can be seen from this figure that the values of $t_{nuc}(\tau)/t_{act}(\tau)$ are, within a good approximation, independent of the initial length of the FR source and independent of the discretization length as long as $l_{max} \leq L_{FR}/10$. We have used these results for the dislocation loop nucleation model of GP-DDD.

The overall procedure for source operation in GP-DDD is as follows. At each time step of a simulation, if the total shear stress acting on the source exceeds activation stress τ_{act} for a period of time $t_{nuc}(\tau)$, a square dislocation loop of area A_i with the same Burgers vector as the original source is introduced centered around the source⁷. In order to determine $t_{nuc}(\tau)$ for a source with a particular length at a given stress, we simply interpolate the 3D-DDD results shown in Fig. A.2(b). When a new loop is introduced, its area A_i is added to the total swept area in the computation of the plastic strain.

⁷ We set A_i equal to $9L_{FR}^2$ which was the approximate loop size emitted during a *ParaDiS* simulation of a FR source.

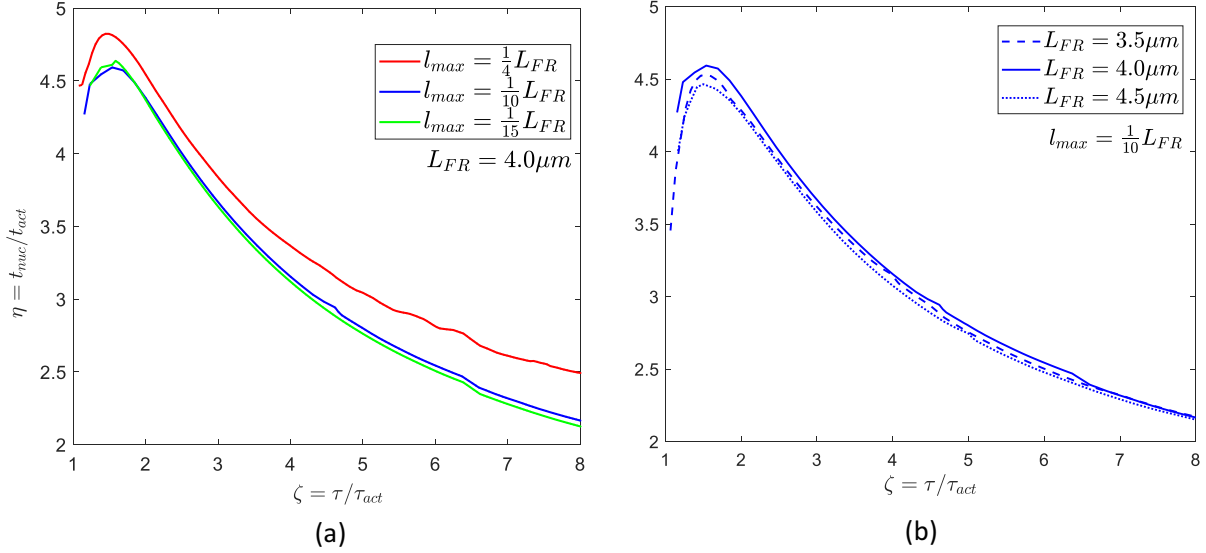


Figure A.2. Values of normalized loop nucleation time vs. normalized stress obtained by 3D-DDD for Frank-Read sources with lengths of $L_{FR} = \{3.5\mu m, 4\mu m, 4.5\mu m\}$, with (a) showing the influence of the discretization length l_{max} and (b) the source length .

Appendix B. Corrective force multipliers for GP-DDD

Driving forces on dislocation segments can be broken down into three contributions: applied external stress, segment-segment elastic interactions, and core energy. By representing a dislocation loop by a rectangle consisting of two screw and two edge segments, we are introducing a geometrical approximation which would affect the plastic strain of the system; this is because rate of expansion of a rectangular loop is not the same as the original loop, when both are subjected to same amount of external stress. For this reason, modifications should be made to the GP-DDD loops to make their area expansion rates as close as possible to the original loops. Denoting the areas of the original loop and GP-DDD loop by A_1 and A_2 respectively, assuming $A_1 = A_2$ at one time step, ideally we should have $dA_1 = dA_2$. For simplification, in the following, the original loop which is roughly in the shape of an oval, is idealized by a circle, and the GP-DDD loop by a square. Correction factors are derived for the case of a square loop to match the amount of swept area by a circular loop. Thus, assuming areas of a square and circle loops, which are denoted by $A_s = a^2$,

and $A_c = \pi r^2$ respectively, are equal to each other, the goal is to make $dA_s = 2\sqrt{A_s}da$ equal to $dA_c = 2\sqrt{\pi A_c}dr$ or equivalently $da = \sqrt{\pi}dr$. This condition translates into $V_s = \frac{\sqrt{\pi}}{2}V_c$, where V_s and V_c are the segment velocity of square loop and nodal velocity of the circular loop, respectively. Thus, $V_s = \frac{\sqrt{\pi}}{2}V_c$ is a necessary and sufficient condition for the square and circular loops with the same area to have the same expansion rate. Each of the three force contributions are investigated separately to ensure the above condition holds.

Applied external stress

When two circular and square dislocation loops with radius and side length of r and a , respectively, are subjected to shear stress of τ^{app} , a force per length of $f = \tau^{app}b$ is exerted on the dislocation segments. Considering that in GP-DDD segments are moved, while in 3D-DDD nodes are displaced, segment and nodal velocities of square and circle loops are equal to

$$\begin{aligned} V_s &= \frac{f_{seg}^S}{B_s} \\ V_c &= \frac{f_{nodal}^C}{B_c} \end{aligned} \tag{B.1}$$

where B is the drag coefficient. For the condition $V_s = \frac{\sqrt{\pi}}{2}V_c$ to hold, and assuming $B_s = B_c$, we should have $f_{seg}^S = \frac{\sqrt{\pi}}{2}f_{nodal}^C$; this is equivalent to using a modification factor of $\frac{\sqrt{\pi}}{2}$ in the case of GP-DDD for contribution of segment forces from applied external stress. Note that using $B_s = \frac{2}{\sqrt{\pi}}B_c$ will have the same effect.

Segment-segment interactions

It can be shown that the elastic energy of square and circular loops is proportional to the square root of their areas, i.e., $E_{el}^S = \alpha_s\sqrt{A_s}$ and $E_{el}^C = \alpha_c\sqrt{A_c}$, where α_s and α_c are parameters that depend on the elastic constants, Burgers vector, and core radius. Considering only this contribution to segment forces, conservation of forces implies that

$$dE_{el} = \frac{\alpha dA}{2\sqrt{A}} = dW = f \cdot dx \cdot p \quad (B.2)$$

where W is the amount of work done on the dislocation loop to move it by dx , dx equals the change in the radius for circular loop i.e., $dx = dr$ but equals $dx = \frac{da}{2}$ for a square loop and p denotes the perimeter of the loop. Substituting for area and its change for the two loops in the expression (B.2) we obtain

$$A_s = a^2 \Rightarrow dA_s = 2ada \Rightarrow f_s = \frac{\alpha_s}{2\sqrt{A_s}}$$

$$A_c = \pi r^2 \Rightarrow dA_c = 2\pi r dr \Rightarrow f_c = \frac{\alpha_c}{2\sqrt{A_c}}$$

Moreover, forces f can be related to velocities using mobility laws as written in relation B.1; thus, assuming $B_s = B_c$ from above relations and using $A_s = A_c$, the ratio of $\frac{V_s}{V_c}$ is obtained as

$$\frac{V_s}{V_c} = \frac{f_s}{f_c} = \frac{\alpha_s}{\alpha_c}$$

Recall that the required condition for the equality of area change is $V_s = \frac{\sqrt{\pi}}{2} V_c$; this condition can be satisfied by using a multiplier of $\frac{\sqrt{\pi}}{2} \frac{\alpha_c}{\alpha_s}$ in front of f_s . This means using a modification factor of $\frac{\sqrt{\pi}}{2} \frac{\alpha_c}{\alpha_s}$ in the case of GP-DDD for the contribution of segment forces from elastic interactions with other dislocations. Lastly, the value of $\frac{\alpha_c}{\alpha_s}$ can be obtained by plotting the elastic energy of circle and square loops versus size, using the non-singular dislocation theory [29], as shown in figure B.1. This yields that $\frac{\alpha_c}{\alpha_s} = 0.912$ for the parameters used here.

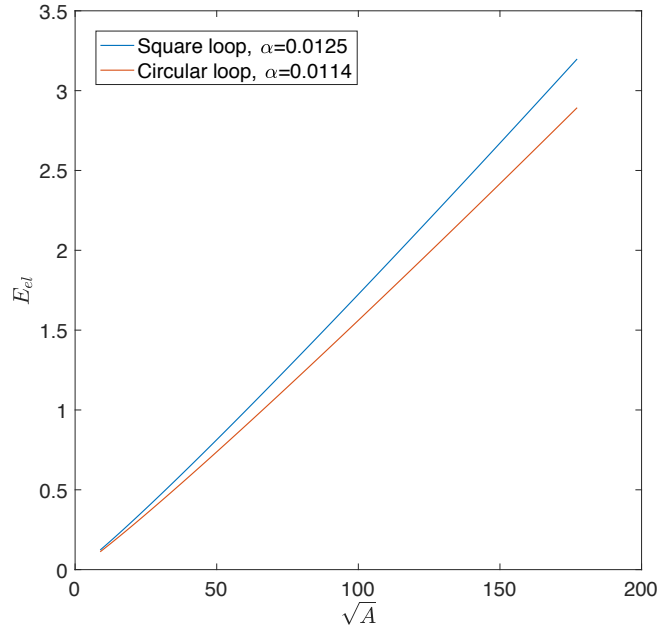


Figure B.1. Elastic energies of circular and square loops for different loop sizes; α is the proportionality constant between E_{el} and \sqrt{A}

This force multiplier corrects for elastic interaction between segments within the same loop. However, there are also interactions between segments within different loops. Given the large number of possible configurations between loops, it is not possible to derive a simple correction factor for these interactions; we assume a correction factor of 1 for these forces.

Core Energy

For this contribution, we simply compute the core energy force per unit length with loop simulations and find $\frac{f_s}{f_c} = 1.13$ (we used the default core energy model in *ParaDiS*). Thus, in order to satisfy the condition that the rates of area change be equal, $V_s = \frac{\sqrt{\pi}}{2} V_c$, f_s should be multiplied by $\frac{f_c \sqrt{\pi}}{f_s 2} = 0.7843$. Thus, a force multiplier of 0.7843 is used in the case of GP-DDD for the core energy portion of segment forces.